DeepScribe: Localization and Classification of Elamite Cuneiform Signs Via Deep Learning

EDWARD C. WILLIAMS, Independent Researcher, USA

GRACE SU, Department of Computer Science, Columbia University, USA

SANDRA R. SCHLOEN, Digital Studies, University of Chicago, USA

MILLER C. PROSSER, Digital Studies, University of Chicago, USA

SUSANNE PAULUS, Oriental Institute, University of Chicago, USA

SANJAY KRISHNAN, Department of Computer Science, University of Chicago, USA

Twenty-five hundred years ago, the "paperwork" of the Achaemenid Empire was recorded on clay tablets. In 1933, archaeologists from the University of Chicago's Oriental Institute (OI) found tens of thousands of these tablets and fragments during the excavation of Persepolis. Many of these tablets have been painstakingly photographed and annotated by expert cuneiformists, and now provide a rich dataset consisting of over 5,000 annotated tablet images and 100,000 cuneiform sign bounding boxes. We leverage this dataset to develop DeepScribe, a modular computer vision pipeline capable of localizing cuneiform signs and providing suggestions for the identity of each sign. We investigate the difficulty of learning subtasks relevant to cuneiform tablet transcription on ground-truth data, finding that a RetinaNet object detector can achieve a localization mAP of 0.78 and a ResNet classifier can achieve a top-5 sign classification accuracy of 0.89. The end-to-end pipeline achieves a top-5 classification accuracy of 0.80. As part of the classification module, DeepScribe groups cuneiform signs into morphological clusters. We consider how this automatic clustering approach differs from the organization of standard, printed sign lists and what we may learn from it. These components, trained individually, are sufficient to produce a system that can analyze photos of cuneiform tablets from the Achaemenid period and provide useful transliteration suggestions to researchers. We evaluate the model's end-to-end performance on locating and classifying signs, providing a roadmap to a linguistically-aware transliteration system, then consider the model's potential utility when applied to other periods of cuneiform writing.

1 INTRODUCTION

Written documents are a central pillar of the study of ancient history. Scholars rely on these primary sources to provide insight into social, political, economic and cultural history. For the ancient Near East and Mediterranean, our historical sources are often those inscriptions that have survived on durable materials like clay. These texts may contain economic records, royal decrees, personal letters, epic poems and stories, among many other types of communication and record-keeping spanning thousands of years of history. This paper considers one corpus of written records of the Achaemenid Empire: a large archive of administrative texts from Persepolis [30] [58] [59] [32]. While the archive, known as the Persepolis Fortification Archive (PFA) preserves texts in Aramaic and a few outliers in Old Persian, Greek, Akkadian, Phrygian, and Demotic, most texts are written in Elamite, a language

isolate written and spoken in Iran in antiquity. It is the primary corpus of cuneiform Elamite texts that serves as the core of this study.

The process of transcribing a cuneiform text is slow and laborious, requiring a great deal of training and expertise. Producing a sign-by-sign transliteration of a text could easily take an experienced scholar multiple days. Many researchers wish also to annotate digital images to indicate the location of a sign on a tablet, whether to use as a pedagogical tool or to more clearly communicate their interpretation. Even a dedicated student worker would find it hard to annotate images of more than a few tablets per day. Due to their terse nature and frequent occurrence, many administrative documents remain untransliterated and untranslated. Assuming archaeologists will continue to discover new cuneiform text corpora, the ability to automatically annotate cuneiform tablet images and to produce acceptably accurate transcriptions would speed up the research process and provide a great boon to historians of the ancient Near East. Our purpose is not to remove the skilled cuneiformist from the process but to spare them the most tedious work.

Current machine learning methods are able to recognize modern handwritten text with accuracy and performance close to that of humans, due to developments in deep neural networks and large-scale datasets with a variety of data instances [16]. Character classification datasets such as MNIST [38] possess tens of thousands of carefully curated training examples, while in-context handwriting recognition datasets such as IAM [45] contain over 1500 scanned documents with line-by-line annotations. However, using a model to automatically transcribe an ancient text is substantially more challenging than typical problems in handwriting recognition. First, the fragmentary condition of these clay tablets leads to inherent data quality problems where even a trained human may not be able to determine what sign a groups of wedges represents. Second, handwriting recognition methods are normally applied to two-dimensional printed or handwritten text on flat surfaces. Because cuneiform signs are produced by impressing a reed stylus into wet clay, the signs are inherently three-dimensional. It is important to note that the tablet surface is also a variable three-dimensional object, making digital photography challenging. A cuneiform text is most easily read when the tablet is held in the hand and rotated dynamically to produce variable light and shadow across the tablet surface. The PFA digital images with annotated hotspots are JPG files of single, statically lit cuneiform texts. Finally, there is simply a lack of comparable large-scale datasets for ancient Near Eastern scripts. Data archives for these scripts exist, such as ORACC [61], which contains thousands of transliterations, and CDLI [19], which contains thousands of cuneiform tablet images , some paired with transliteration. However, these archives do not possess sign-level bounding box annotations that enable the straightforward use of modern object detection methods.

The goal of this paper is to present a new machine learning pipeline that leverages state-of-the-art results in computer vision with a large, curated corpus of sign annotations to train computer vision models to localize and classify signs on images of cuneiform tablets. Prior work on automatic cuneiform transcription addresses the lack of data by generating artificial training examples [53] or utilizing weakly-supervised learning [20]. The former method generates training data for a sliding-window object detector using hand-drawn cuneiform autographs, but is limited by the size and scope of the autograph dataset. The latter work uses line-by-line transliterations to train an object detector, but observes that fully-supervised training using a small corpus of manually annotated tablets provides a significant boost to test performance. Thus, reliable automated cuneiform transcription remains an open challenge, one which we anticipate will be aided by a large training corpus.

To this end, we contribute a novel use of the richly annotated Persepolis Fortification Archive - training data for cuneiform sign localization and classification models. The PFA editorial team has produced hundreds of thousands of digital images of tablets, many of which have been annotated with per-sign bounding boxes, referred to as hotspots, and labeling information. We extract PFA text and image data with annotations from the Online Cultural and Historical Research Environment (OCHRE) data repository and convert it to the standardized machine-friendly representation used in the Detectron2 object detection library [66]. To improve the quality of the dataset, we performed an additional round of manual refinement of the bounding box annotations, removing

low-quality (blurry or poorly lit) images from the dataset, and relabeling numeric signs to ensure consistency with non-numeric sign labels. We analyze the distribution of labels in the dataset, finding a power-law distribution of label frequencies. While this is fairly typical of natural language datasets, it does imply that care will need to be taken to properly evaluate predictive performance on lower-frequency signs.

We then use this dataset to construct a modular computer vision system for automating the first steps of cuneiform tablet annotation. We use a RetinaNet [40] object detector to identify rectangular image regions containing a single cuneiform sign. This is modeled as a single-class object detection task, where we leave the problem of identifying the specific sign to a dedicated classification network. We find that this object detector can achieve a localization Average Precision at 50% overlap (AP@50) of 0.78 across held-out test images, corresponding to a qualitatively effective sign detector. We perform ablation studies on both classification and bounding box detection tasks to estimate the value of additional labeled training examples, finding that the relative value of new training examples decreases as dataset size increases.

The identity of a sign is determined by a 141-class image classifier trained on ground-truth hotspot regions. We first evaluate this classifier on held-out ground-truth hotspot regions to assess the difficulty of the recognition task, finding an average top-5 held-out accuracy of 0.89. We observe a degradation in top-5 accuracy to 0.8 when performing inference on hotspot regions predicted by the pre-trained detector. Finally, hotspot locations are sorted into discrete text lines using a variant of the Sequential RANSAC algorithm [63]. End-to-end systems similar to this exist [51], but we chose to design our pipeline with modularity in mind, inspired by other work in computationally-efficient text localization [16]. Sub-components from diverse data sources could then be fine-tuned or retrained separately. However, we find that the sequence reconstruction accuracy as measured by Character Error Rate (CER) is relatively poor, achieving a CER of 0.69. While the hotspot detector and sign classifier in concert are able to provide useful suggestions to cuneiformists, a fully end-to-end transliteration system will likely require explicit modeling of linguistic context. We outline a path forward to incorporate linguistic information in future work to eventually produce complete tablet transliterations from an unannotated image.

2 BACKGROUND

The automated annotation of historical documents is of keen interest to digital humanists, archivists, and archaeologists. A fully digitized corpus of documents would enable the rapid search and analysis of historical archives, with the ability to collect primary sources for historical research far more quickly than manually clicking through and inspecting a set of images. Beyond the implications for document retrieval, digitized historical information can enable innovative applications of statistical analysis tools such as network analysis [25][22][2] [3], reconstruction of fragmented documents [1] [4], and the use of modern natural language processing tools to aid in translation or even decipherment. [43] [57] [44] [55] [54] [28]

Tools to partially automate or aid human annotation would enable the rapid creation of richly annotated digital archives. The success of machine learning algorithms in image analysis tasks both inside and outside of document recognition [37] [68] [40] [51] [16] [33] and the proliferation of open-source software tools for developing machine learning products [11][66][47] present the possibility of using these tools for the analysis of historical documents. However, historical document archives pose unique problems to the application of machine learning—the first of which is the generation of training data, which, for document annotation, can only be described as a "chicken and egg" problem. Documents must be annotated in great quantities to produce training data for most machine learning algorithms, which, of course, assumes that a researcher already has annotated most of the documents that they may be interested in annotating. This has led to approaches that, much like ours, seek to leverage existing datasets to improve the annotation of future documents [50] [21] [18] as well as using machine learning techniques intended for low-data regimes [20].

Fig. 1. PFA Tablet Image with Annotated Hotspots viewed in OCHRE

2.1 The Persepolis Fortification Archive

In 1933, the Oriental Institute Expedition to Persia discovered a large cache of cuneiform tablets at the site of Persepolis. The tablets were discovered in two rooms inside a fortification wall and were dubbed the Persepolis Fortification Archive (PFA) [59]. Scholars at the Oriental Institute and world-wide have been studying and publishing editions of these tablets since 1937 [30]. The tablets provide valuable insight into the administration and economic history of the Achaemenid Empire, and their digitization enables the rapid search and indexing of this large corpus. The majority of the PFA texts record the distribution of commodities and allocation of resources, which has led to a greater understanding of the internal administrative details structure of Persia [32]. In 2006, a renewed effort to document and publish the texts led to the creation of the PFA dataset as it exists today. UChicago professor Matthew W. Stolper assembled an international editorial and technology team to work on the PFA project. The team included specialists in Elamite and Aramaic language, seal iconography, digital imaging, and data representation. The editorial team worked on new editions of the texts. The seal iconography team faced the massive task of identifying, classifying, and studying the thousands of unique seal impressions recorded on the tablets [27]. The photography team instantiated three separate digital photography modalities: conventional digital photography, high-resolution scanning under filtered light, and Polynomial Texture Mapping (also known as Reflectance Transformation Imaging) [58]. The database team consulted with the rest of the PFA team to record all of this information in an integrated system called the OCHRE database platform.

2.2 The OCHRE Platform

The Online Cultural and Historical Research Environment (ochre.uchicago.edu) is a research database platform created to meet the unique demands associated with research data in the humanities and social sciences. Data in OCHRE is stored in highly atomized and integrated documents. For the PFA data, each clay tablet, each digital photograph, and even each cuneiform sign in each text is a discrete database object, described by metadata and associated with hundreds or thousands of other database items. The PFA project has used OCHRE for over a

decade to pursue a variety of research goals, all related to the study of the information recorded on the clay tablets.

In OCHRE, the PFA editorial team creates text editions in which each cuneiform sign is recorded as a discrete database item. Each sign reading represents an interpretation by a PFA editor reached through careful examination of the clay tablet under magnification which leads to a reading of the signs and translation. As shown in Figure 1, PFA team members use OCHRE to associate manually-created tablet image hotspots with transliteration values of the text in question, thereby integrating the digital image with the text transliteration. Linguistic information is therefore incorporated at the sign annotation level even if individual signs may be difficult for humans to read in isolation. The original intent of the hot-spotting effort was to create a pedagogical tool to help teach the reading of Elamite. Instead of providing students with a hand-drawn version of a cuneiform text with highly regularized cuneiform signs—as it is often the case in the classroom—these annotated images would allow the student to engage with the cuneiform text as recorded by the scribe. During the early discussions that led to this project, we realized that this carefully curated dataset of over 100,000 labeled hotspots could also serve as a training set for an automated sign detection and identification tool.

To account for the polyvalent nature of the logographic and syllabic cuneiform writing system, each sign value links to a database item that represents the sign. For example, the sign GIŠ can communicate syllabic values *is/ez*, *giš* and *bil* as well as the logographic usage meaning "wood" and the determinative usage which refers to wood and wood-based products. The identification of the value of a sign must be determined by context. In this stage of the DeepScribe project, we set aside the sign–to–value mapping problem and focus entirely on sign identity prediction. In future work, we plan to leverage the extensive PFA glossary of Elamite words in OCHRE to suggest sign values and word identifications.

2.3 Related Work

Extensive research on computer vision methods for document recognition has been conducted over the past 30 years, resulting in software packages that are widely used in both research and industry [56] [39]. Such methods rely on software pipelines that identify lines of text, segment lines into words or characters, and identify characters. Handwritten text recognition, due to the relative irregularity of handwritten characters, typically requires dedicated processing pipelines [16] [52] [29] [33] but can be performed effectively by an automated system, with character error rates under 8% [5] Full-page handwriting recognition methods typically segment an image into text lines via object detection methods that are then decoded using an alignment-based loss function [65] [16], although segmentation-free methods exist and are known to be effective [5]. We apply a similar factorization, first using an object detector to identify cuneiform signs and then to classify them, although our access to per-character bounding boxes allows us to avoid the problem of line segmentation.

The success of these methods has inspired their application to historical document analysis, developing tools to automatically extract useful semantic information from documents that are typically only readable by specialized scholars. [18] describes a text recognition system intended for use on handwritten historical documents, as well as a publicly available software tool that allows for fine-tuning on user-provided corpora. [9] provides an excellent review of projects applying techniques from computer vision to cuneiform tablet images, as well as a typology of the diverse methodologies that have been adopted to analyze these data sets. While the relatively small datasets of 3D-scanned or photographed cuneiform script typically limit the utility of the large deep learning models that are often successful in addressing computer vision problems, digital humanists and computer scientists have spent decades building automated systems that can leverage what data exists in archives such as the Cuneiform Digital Library Initiative and ORACC. [7] uses a keypoint-based method to identify cuneiform signs on vector graphics representations of cuneiform text, but this requires the manual creation of a vectorized transcript before any analysis can be performed.

The recent work of [20] uses a weakly-supervised algorithm to learn to detect cuneiform signs from a 2D image dataset of Neo-Assyrian tablets found on ORACC [61], which are linguistically distinct from our Elamite tablet set. Due to the lack of annotated sign localizations in the ORACC dataset, they develop an iterative learning procedure using a Single-Shot Detector (SSD) object detector algorithm [42] along with line alignment techniques to iteratively propose, filter, and retrain on automatically generated sign localizations. Additionally, this work includes a Conditional Random Field (CRF) sequence-modeling step as the final stage of the annotation-alignment pipeline, but this uses geometric and spatial information to match predicted and classified regions to transliteration characters, rather than using contextual information to refine the raw predictions of a sign detector. They also find a boost in accuracy when their models are provided with a small set (< 60) of fully annotated tablet images. We adopt a similar framing of the problem of cuneiform sign localization as object detection, but are able to directly train on localized sign annotations rather than resorting to a weakly-supervised approach. However, we see several promising avenues for future works in the combination of these two approaches, such as the leveraging of a model trained on a large dataset of annotated signs to initialize weakly-supervised learning methods on low-resource dataset, although there are many questions to be answered on the efficacy of cross-linguistic transfer.

An orthogonal approach that that may be better suited to text written on curved tablets uses 3D scanning information as input to cuneiform tablet analysis. However, acquiring 3D-scanned data requires far more time, labor, and cost than 2D image data. [6] uses 3D models of cuneiform tablets to extract vector drawings and retrieve the corresponding cuneiform characters. [8] also uses 3D data, but predicts the time period of a cuneiform tablet instead by training a Convolutional Neural Network (CNN) on a partially transliterated cuneiform tablet 3D surface mesh dataset. [36] addresses the geometric nature of cuneiform script by classifying signs based on graph model data of cuneiform signs extracted from 3D scans using the method of [24], but still must rely on a small set of annotated data. [53] attempts to resolve this by generating artificial 2D projections of 3D data using a Generative Adversarial Network (GAN), although annotations still must be provided manually for downstream classification tasks.

3 DATA

The annotated Persepolis Fortification Archive tablet photos present an ideal dataset for the training of machine learning models for cuneiform character localization and recognition. Unlike other labeled cuneiform datasets, which typically consist of whole-tablet images paired with transliterations [61], the PFA tablet images in OCHRE are labeled much like object recognition datasets such as COCO (Common Objects in Context) [41]. Each image of a PFA tablet is paired with a set of labeled bounding boxes, which can either be used out-of-context in image classification tasks, or in-context for bounding box detection tasks. We explore both in this work as part of a modular recognition pipeline. Furthermore, the dataset is composed of images of tablets themselves, not line drawings or artificial constructs - the dataset even contains natural handwriting variation among scribes. This allows computational models to learn directly from cuneiform text as it was written by scribes. We anticipate that this will produce models capable of recognizing cuneiform text *in situ*, without requiring human scholars to perform tedious pre-processing before annotations can be generated.

3.1 Descriptive Statistics

The relative frequencies of the 141 unique Elamite signs contained within the dataset are highly unbalanced, with the vast majority of signs being fairly poorly attested. Power-law behavior of natural language datasets rank-frequency relations has been attested across languages and corpora. Compliance with this behavior, referred to as Zipf's Law [48], is most simply checked by fitting the log-log rank-frequency data of a corpus to a line (although this method has some error modes, see [17]). Interestingly, the rank-frequency distribution of the PFA sign data is not well fit by a single power-law ($r^2 = 0.66$ but poor visual fit), but is well described by a broken

Fig. 2. Histogram of Sign Frequencies

Fig. 3. Log-Log Rank-Frequency Plot

power law ($r^2 = 0.98$, see Figure 3), appearing as a piece-wise linear fit in the log-log plots. This behavior is not unexpected from a modern natural language corpus [49]. We observe that the PFA corpus exhibits a steep "fall off" in the usage of all available signs. The scribes writing Elamite used a script system which developed 2,700 years earlier for Sumerian, an unrelated language. While scribes of the Elamite language employed a common subset of signs, they also had at their disposal a wide range of rare signs, including syllables and logograms.

This primarily implies that care will need to be taken to evaluate performance. High performance on high-frequency signs may "wash out" low performance on low-frequency signs on aggregate metrics. As would be expected from a power-law distributed data set, the top 50 most frequent signs cover 86.1% of the dataset, indicating that the remaining majority of sign classes are poorly represented. As such, there may be a trade-off between a system focused on low-resource signs and a system capable of automatically annotating "easy" or high-resource signs then flagging others for human intervention. We explore the relationship between class frequency and performance in Section 5.2.1.

3.2 Preprocessing

For further analysis and machine learning experiments, the PFA OCHRE dataset was exported into the JSON file format utilized by the Detectron2 [66] object detection library. The dataset consists of 5007 images of 1360 unique tablets, where each image contains a set of hotspot region annotations. Bounded hotspot regions (equivalent to bounding boxes in an object detection framework) are annotated with a categorical label corresponding to the Elamite sign contained within the hotspot.

Because tablets were photographed from various angles, not every sign in every image is annotated. To remove large unlabeled regions present in many PFA images, typically consisting of unannotated cuneiform signs, we automatically cropped the tablet images using the pixel locations of the tablet's bounding boxes. Hotspot locations were then adjusted to be consistent with these modified dimensions.

After early experiments were performed, we discovered that a subset of data was too noisy to contribute to the training. We culled images with very low resolution, with signs out of focus or located on an extreme edge of the tablet, all of which contributed to making the sign illegible even to the human eye. Taken as a whole, however, the level of noise in the dataset was quite low.

3.3 Data Anomalies

3.3.1 Cuneiform Numerals. The PFA project had not identified the cuneiform signs used to record numerals, requiring some intervention in correcting these sign labels. In the cuneiform writing system used by Elamite, numbers are expressed with a a combination of wedges. For example, the number "11" is represented by the concatenated signs for "10" and "1". Each of the elements corresponds also to a sign with a syllabic value. 10 is represented by the sign U and 1 is represented by the sign DIŠ. In the traditional transliterating system also employed by PFA, numerals are transliterated with Arabic numerals, making it sometimes difficult to identify the original cuneiform signs. To avoid mistakes in the recognition of numbers (1 recognized as DIŠ and vice versa), we merged all simple numerals with their syllabic value 1 = DIŠ and 10 = U. We also removed composite numerals from the dataset, for example 90 (DIŠ + UUU) or 11 (U + DIŠ).

3.3.2 Broken/Corrupted Signs. While the majority of the hotspotted PFA tablets are in good condition, some tablets have suffered damage in the roughly 2500 years in between their creation and their recovery by the OI. Damage in this corpus takes the typical forms observable in other clay tablet corpora: abraded surfaces, prominent cracks, and broken edges or sections of the tablet. In practice, this leads to fairly predictable inference errors—for instance, as discussed in Section 5.1.1, we observe some false positive detector predictions that consist of misidentifying cracks in a tablet as cuneiform signs.

3.4 Public Distribution

A subset of the dataset is available in JSON format [1]. Some of the images produced by the Persepolis Fortification Archive document texts are not yet published by the editorial team. Only images of previously published texts are included in the subset provided at the link above.

3.5 Metrics and Evaluation

Given the novelty of this dataset in the context of computer vision, we use a series of metrics designed to evaluate model performance on subtasks within our dataset. Before constructing any end-to-end system, we would like to know, primarily, how easily an object detector can localize *any* cuneiform sign, regardless of identity, and how well an off-the-shelf classification algorithm can recognize cuneiform signs when provided with "correct" (i.e. human-generated) image regions. For the former task, we rely on standard object detection metrics, prioritizing the single-class Average Precision (AP) used to evaluate object detectors on other datasets [40]. This metric is computed with respect to an Intersection over Union (IoU) threshold that demarcates whether a prediction is a true or false positive. Per-class AP values are computed, and then averaged over all classes (in the multi-class case). These values can be computed over varying IoU thresholds, but we focus on the AP@50 (50% IoU overlap) metric. State of the art multi-class object detectors can achieve AP@50 metrics upwards of 61% on large datasets such as COCO [40].

For sign classification, we focus on the top-k accuracy [2], while also using metrics robust to dataset imbalance such as the mean recall [3] across all classes. In early discussions with potential users of this tool, providing a list of ranked suggestions for each sign on a new document was identified as a useful feature, and the top-k accuracy provides a useful measure of success on this task. Given the large dataset imbalance we observe in Section 3.1, we also wished to characterize classification models' performance as training set attestations vary, to understand which sign classes may be particularly difficult to recognize even in the context of "perfect" localizations. For context, deep image classifiers trained on clean, multi-class datasets such as ImageNet can achieve top-1 accuracies upwards of 79%, and top-5 accuracies upwards of 95% [31] on datasets with far more classes.

---

[1]see http://github.com/edwardclem/deepscribe for download instructions and source code.
[2]Where a prediction is marked as a true positive if the correct sign is in the top k of a ranked list of predictions.
[3]In this context, "recall" refers to the fraction of true positives that are flagged as such.

Fig. 4. DeepScribe Vision Pipeline

We evaluate end-to-end transliteration performance in terms of the Character Error Rate (CER), defined as the edit distance between the predicted and true sequences normalized by length. While subtask models may be useful on their own, this metric will determine how well an automated system can produce a whole-document transliteration without human intervention. Modular transliteration systems trained on the 1500-document IAM corpus can achieve CERs under 10 percent [16], indicating a high bar for effectiveness on this end-to-end task.

4 METHODS

The structure of our modular computer vision pipeline is designed to explore two sets of questions—both to understand performance on the individual "subtasks" presented by our dataset, as described in section 3.5, as well as to combine individually trained components to produce an end-to-end OCR system. While it is certainly possible to model this dataset using end-to-end object recognition systems (possibly with an additional linguistic refinement component), the modular structure of this workflow arose from our exploration of individual tasks that could be accomplished within the dataset. Training modularized subcomponents of OCR systems has precedent in the transliteration of modern texts [16], but the primary purpose it serves here is to improve our understanding of the novel PFA dataset from the perspective of computer vision. Modeling OCR as a combination of object recognition and transliteration tasks is fairly common in the field [65] [16], with one key difference—outside of prior work in cuneiform tablet transliteration [20], it is relatively rare to perform OCR on the word or character level for Latin scripts. However, prior work transliterating Chinese characters [67] has adopted a per-character object recognition approach, showing that different linguistic contexts often require customized pipelines.

4.1 Pipeline

Our vision pipeline consists of three stages: hotspot detection, sign classification, and line detection (Figure 4). Rather than jointly training a model to produce hotspots and classify signs, we decided to split the problem into two independent modeling steps, inspired by [16]. As such, each stage in the pipeline is trained separately using ground-truth annotations. Potential users of this modeling pipeline identified being able to manually adjust suggested hotspot regions as a key feature, which was facilitated most easily by suggesting hotspots first and then performing inference using a different network. An end-to-end workflow that also incorporates user feedback at the hotspot stage could possibly be accomplished by a two-stage detector such as Faster R-CNN [51]. In fact, our architecture is similar to training a proposal generator and classification network separately. We also wished to perform separate analyses of learning tasks within the PFA dataset. These serve to characterize the

"difficulty" of each learning task (i.e. hotspot localization, sign-classification) given the novelty of this dataset for machine learning training. There may in fact be synergies that an end-to-end system could exploit, such as using contextual information to refine classification prediction, although these could also be achieved using a final, modularized language modeling step.

4.2 Sign Detector

We define cuneiform sign localization as an object detection problem - i.e. selecting rectangular bounding boxes that contain a single cuneiform character. All boxes are assigned to an identical class and an object detection algorithm is trained to distinguish the class against background regions. We use a RetinaNet [40] as implemented in the Detectron2 library [66] to perform sign localization. The RetinaNet was trained using the Adam optimizer, with an initial learning rate of 1e-4 and a learning rate scheduler that decreased learning rate when performance on validation loss plateaued. Before training, images were rescaled using preset per-channel mean and standard deviation values originally defined on ImageNet. Images were augmented using a mixture of image rescaling, random cropping, and random horizontal flipping [4] and fed into the network in batches of 10. The network was limited to a maximum of 40000 iterations, with early stopping performed once the model's validation performance had ceased improving. Final performance metrics on validation folds were computed using the best iteration as judged by AP@50.

4.3 Sign Classifier

We use a ResNet image classifier to map tablet image regions containing a single sign to a sign class. The classifier maps an image to a 141-dimensional vector of logits, which we use to rank predictions and compute top-k results. Image regions were resized and padded to 50 x 50 pixels before input into the network. During initial experiments, we found that data augmentation (RandomAffine, ColorJitter, and RandomPerspective as implemented in torchvision) as well as standard BatchNorm layers are effective at preventing overfitting, and thus do not apply any kind of weight decay. The classifier network was trained using the Adam optimizer, with initial learning rate 1e-3 for a maximum 500 epochs across all experiments. A learning rate schedule with a patience of 5 epochs was used to reduce learning rate by a factor of 10 when validation loss plateaus. After 10 epochs of no improvement, learning was terminated. The classifier was implemented in PyTorch [47] using the TIMM library [64] of predefined model architectures. Statistics were computed at the final epoch of training. Interestingly, we found that using pre-trained models (obviously trained on a very different domain) improved convergence time but not validation performance. To remove unexpected confounding variables, we perform all analyses in this paper on randomly initialized models. We train all classification models on ground-truth, human-annotated hotspot regions and perform tests on both held-out ground-truth hotspots and hotspots predicted by the object detector described in Section 4.2.

We experimented with several methods to address severe class imbalance during classifier training, including reweighting per-class loss using inverse frequencies [35], reweighting with inverse log-frequencies, and using the focal loss [40] that was also used to train the RetinaNet sign detector. However, results were inconclusive (See Appendix C) and we perform end-to-end inference using a classifier trained using the unweighted cross-entropy loss.

4.4 Line Detector

To sort detected hotspots into discrete "lines" facilitating the left-to-right reading of Elamite texts, we use a variant of the Sequential RANSAC algorithm [63] to iteratively assign elements the set of hotspot centroids on a

---

[4]Flipping transforms, especially vertical flipping, may be problematic in the case of cuneiform characters, many of which are not invariant to rotations or reflections. However, our detector is designed to recognize any character as a general class, and not predict identity.

tablet to a set of linear models. The algorithm fits a series of L2-regularized RANSAC linear models to the data. Regularization serves to produce lines as flat as possible, and we force each linear model to have a slope less than 0.3 as a further constraint. These parameters, tuned manually, were found to encode our prior belief that lines in this corpus are generally flat or very slightly slanted and rarely intersect. Outliers from the initial RANSAC procedure are used to fit additional RANSAC linear models until there are one or fewer points remaining. Final outliers are assigned to their nearest text line by Euclidean distance. The set of text lines are sorted by their y-intercepts to provide a full left-to-right reading order across multiple horizontal lines. Note that this method of sorting could be problematic for highly slanted lines, which are relatively rare in cuneiform texts, but in practice the majority of the text lines in the PFA are flat enough to avoid significant issues.

4.5 Evaluation Strategy

We perform by-tablet cross-validation splits, instead of splitting our data randomly by image. Some images contain different regions of the same tablet, and we wish to estimate the performance of our models on entirely unseen tablets, as that most accurately simulates the scenario of performing inference on a novel cuneiform text. We produce 5 folds, each containing 272 tablets, and one remaining fold containing 271 tablets. Folds may contain slightly different numbers of images and hotspots, due to differences in text length. The folds were kept consistent across all steps of the modeling pipeline—i.e. the sign classifier and hotspot detector were trained and tested on the same sets of ground-truth hotspots. This ensures that when the sign classifier is used to perform inference on predicted hotspots, the hotspots are unseen by both stages of the modeling pipeline. We then provide metrics aggregated over held-out folds to estimate pipeline performance. We note that when early stopping was performed, held-out fold metrics were used to determine stopping criteria, meaning there may be slight information leakage from the test fold into the fitting procedure. A model evaluation scheme such as nested cross-validation [14] would provide more robust error estimates, but we found this to be too computationally expensive for our initial experiments. However, we note that ablation experiments that involve randomly sampling subsets of the data were performed by image, not by tablet. This produces slightly less robust estimates of out-of-sample performance.

5 RESULTS

We first perform individual analyses of each stage in the pipeline as trained and tested on ground-truth data, and then perform end-to-end inference and evaluation.

5.1 Sign Detector

Cross-validation results of the RetinaNet's single-class object detection performance can be found in Table 1. We observe relatively consistent behavior across folds, indicating that each fold provides sufficient coverage of the sign dataset for the purposes of this training task. We also find that 101-layer backbones slightly underperform the 50- and 18-layer backbone networks, likely due to underfitting. Given that the difference in performance between a 18- and 50-layer backbone is relatively small, it is unlikely although possible that a deeper ResNet may provide a performance benefit. We experiment with using the ResNet18 and ResNet50 backbones for end-to-end experiments in Section 5.2.4.

5.1.1 Qualitative Analysis. We provide an example of predicted hotspot locations along with their ground-truth counterparts, to illustrate the strengths and weaknesses of our trained sign detection models.

This prediction example displays some of the common pathologies exhibited by the detector—difficulty with oblique signs and false positives (although they appear to be partial signs that are not included in the image), and

---

[5]Here we compute the average of per-tablet recalls at a fixed classification threshold, given the provided IoU threshold,

Table 1. RetinaNet Detection Performance

Backbone	AP@50	AP@75	Recall@50 [5]	Recall@75
ResNet18	77.4 (2.3)	20.7 (0.8)	84.4 (1.6)	38.3 (1.5)
ResNet50	77.1 (1.5)	20.5 (1.5)	84.1 (1.7)	37.7 (1.0)
ResNet101	74.4 (3.0)	12.0 (1.9)	82.8 (1.6)	35.5 (1.7)

Fig. 5. Detector Predictions

Fig. 6. Hotspot Locations

overlapping boxes. Other common mistakes we observe are the joining of adjacent signs into a compound sign [6], and the splitting of compound signs into individual bounding boxes. We hypothesize that integrating linguistic context into the detection stage may reduce these sorts of errors, although that may reduce the modularity of the detection module (i.e. limit it to a language such as Elamite where we have a particularly well-annotated dataset).

We observe in other images that the detector struggles with signs on edges or sides of tablets, particularly when they are angled away from the camera. This is likely a consequence of using a 2-dimensional representation of what is ultimately a 3-dimensional object—cuneiform tablets often have signs on their (often curved) sides.We hypothesize that the mis-identification of darkened cracks as signs is due to both the similarity of some thin cracks to simple signs and the appearance of cracked signs in training data.

5.1.2 Ablation. To investigate the effect of dataset size on detection model performance we create a single randomized train/test split and randomly subsample the training set at specified fractions. We then evaluate performance on the fixed test split to estimate the marginal utility of adding additional data points to the training set. We note that this experiment was performed before the numeral label adjustments in Section 3.3.1, and not re-run due to resource constraints. However, the bounding box dataset was only slightly modified when performing relabeling, so we do not expect significant changes to the trend.

A plot of detector AP@50 as a function of dataset size can be found in Figure 7. Two patterns are apparent, the first of which is that a detector trained on a small fraction of the PFA dataset performs surprisingly well by this metric. This suggests that lower-quality but reasonable hotspot detectors can be trained using datasets on the order of hundreds of images. It may also imply that a detector can be fine tuned on a related dataset (i.e. another cuneiform tablet set) with relatively few labeled examples, a topic we look forward to exploring in future work. The second is that while AP@50 increases as more data is added, the marginal utility of a new annotated image diminishes as the dataset size increases. [60] and [34] suggest that there may be a logarithmic

---

[6]Signs that consists of independent subunits but are read as a single sign. These are relatively rare in Elamite, but an example is the compound sign IA which is a concatenation of the signs for I and A. Theoretically, every sign is a compound of the basic wedges "horizontals," "verticals,", "diagonals" and an angular hooked wedge known as a "Winkelhaken."

Fig. 7. Dataset Ablation - Detector

relationship between the size of a large computer vision dataset used to train a high-capacity convolutional neural network model and its test performance. While our dataset is certainly not on the scale of current computer vision benchmark datasets, we observe a similar trend. These results jointly indicate that a RetinaNet object detector is able to produce high-quality segmentations of cuneiform tablets, but that further improvement of the detector is likely bottlenecked by data quality and fundamental limitations of 2D image representations of 3D objects.

5.2 Sign Classification

We report sign classification results on ground-truth annotated hotspots in Table 2, and observe that the the top-1, top-3, and top-5 accuracies of ResNet-based classifier models are relatively consistent across folds. We see an increase in classification accuracy as the depth of the ResNet model increases, although the change between an 18-layer ResNet and a 50-layer ResNet is much smaller than that between a 50- and 101-layer ResNet. We also investigate the relationship between a sign's frequency in the training set and its per-class test performance in Table 3. We find that the mean recall, sometimes referred to as the balanced accuracy [46], increases with network depth, although in a diminishing manner. However, we find that the rank-order correlation between the per-class test recall and the class's train frequency is very high. Similarly, per-class test precision is correlated with the class's frequency in training data. This is unsurprising given the extreme imbalance in class frequencies, and provides a limit to the utility of the models—they are likely to be be more useful in automating the annotation of well-attested signs than predicting results for rarer signs. We provide a more detailed exploration of learned sign class representations in Section 5.2.3. We also hypothesize that providing sign ranking information could improve the performance of a language model attempting to predict the value of a given sign in context, but we leave this exploration for future work.

5.2.1 Qualitative Error Analysis. Figures 8 and 9 contain randomly sampled test-time failure modes of a trained sign classifier on ground-truth annotated hotspots. We point out two types of errors: one where the classifier predicted the top-1 sign incorrectly but the correct sign was within the top-5 predictions, and another where none of the top-5 predicted signs contained the correct sign. A qualitative review of these failure modes indicates that these signs generally would be difficult even for a cuneiformist to identify in isolation. Some signs are nearly impossible to read on photos as they are distorted by the curvature of the writing surface. Others are

Table 2. Model Architecture - Accuracy

Architecture	Top-1	Top-3	Top-5
ResNet18	0.657 (0.003)	0.837 (0.003)	0.888 (0.003)
ResNet50	0.683 (0.006)	0.851 (0.006)	0.897 (0.004)
ResNet101	0.692 (0.005)	0.859 (0.005)	0.902 (0.004)

Table 3. Model Architecture - Frequency vs Performance

Architecture	Mean Recall	Precision ρ	Recall ρ
ResNet18	0.459 (0.013)	0.503 (0.027)	0.773 (0.040)
ResNet50	0.483 (0.018)	0.509 (0.049)	0.794 (0.042)
ResNet101	0.504 (0.022)	0.519 (0.028)	0.763 (0.039)

Fig. 8. Incorrect Top-1, Correct Top-5

Fig. 9. Incorrect Top-5

unrecognizable or hardly recognizable because the sign is damaged, or not all wedges are visible due to poor exposure. Unclear sign borders can lead to misinterpretation, as the sign spacing is highly variable in these texts. The obvious implications of this review are twofold: one, that these classification methods are confused by genuinely difficult cases, and two, that these methods may be useful for automatically flagging ambiguous signs or bad annotations.

5.2.2 Per-Class Performance. We perform a more detailed analysis of the model's performance on a per-class basis using one test fold as an exemplar. Results for the remaining 4 folds can be found in Appendix D. The class distribution of sign images in the PFA dataset is highly imbalanced, implying that performance may be highly variable across classes, and that aggregate top-k accuracy estimates may be biased by good performance on highly

represented signs. This information, in addition to the class rankings discussed above, would be useful to inform users about the expected accuracy of a prediction. We hypothesized that classes that were better represented in the training data were likely to have improved precision and recall, given that high-capacity image classification networks tend to improve in performance (albeit logarithmically) when more training data is provided [60].

Fig. 10. Per-Class Precision and Recall

The relationship between number of training examples and test recall is roughly log-linear (Figure 10), and a class' rank in training example counts is highly predictive of its rank in test recall. The log-linear relationship implies that the classifier benefits from additional training examples but the marginal utility of each training example decreases as the training size gets larger. There are also several signs with extremely low, near-zero recall. A few low-frequency signs have very high precisions, and training dataset frequency is less predictive of test precision.

5.2.3 Ablation. To investigate the effect of dataset size on classification model performance, we also varied the size of the training dataset while keeping the size of the test data constant. To perform these experiments, we used a single train/test split and randomly subsampled the training data without explicit stratification. We performed 10 subsamples for each sample fraction. Figure 11 shows test set accuracy as a function of training set sample fraction.

As alluded to in Section 5.2.1 above, we see a similar pattern in aggregate accuracy as we do with per-class recall—performance increases as more training examples are added, but the rate of improvement slows down as the dataset gets larger. We also see that with a very small training sample the number of classes that are extremely poorly represented (i.e. near-zero recall on test) is high. These results are similar to our observations in Section 5.2.1, indicating a roughly log-linear relationship between sign attestation and predictive performance.

These results, in combination with our qualitative exploration of error modes, imply that precisely disambiguating signs is fairly difficult for a classification model due to a combination of label noise, data imbalance, and linguistic ambiguity. Many signs are difficult to interpret for a human with full access to linguistic context, rendering them even more difficult for a model classifying single signs. We see this show up as well in the large

Fig. 11. Dataset Ablation - Classifier

gap between top-1 and top-5 accuracies — the model clearly is learning representations that enables meaningful semantic grouping of signs, but we may be hitting a limit of single-sign image classification performance on this type of data. Further dataset refinement to remove unclear signs may aid in this slightly, but we suspect that the models are bottlenecked by image quality and the fundamental difficulty of the task.

Fig. 12. Visualization of tSNE Embedding Space.

5.2.4 Analysis of Learned Sign Representations. Cuneiformists traditionally organize sign lists following an artificially established principle first developed for the Elamite script in 1848. The wedge-shaped elements of signs are analyzed from left to right in a hierarchical fashion—horizontal wedges before diagonals, Winkelhaken, and vertical wedges and one element before stacks of two, three and so on [10]. This method is far from ideal as it focuses only on the beginning of the signs and not the overall shape. In contrast, some antique scribes organized their sign lists following the overall shape of the sign (acrographic principle) and typical sign clusters which can appear at the beginning, middle or end of a sign [23]. Given that the question of specifically how to organize and group cuneiform signs has great practical relevance for how cuneiform is taught and understood, we want to explore how algorithmically learned representations of signs are clustered.

To visualize and analyze the sign representations learned by a trained sign classifier, we produce t-SNE plots utilizing a nonlinear dimensionality reduction technique well-suited for visualizing high-dimensional data [62]. Using a trained sign classifier, we embed hotspot images into 141-dimensional logit vectors, encoding the model's unnormalized beliefs about the sign's class. Then, we perform PCA (Principal Component Analysis) on the full sign classifier dataset (including training data as well as the held-out validation data) to represent the dataset in

terms of the top 50 principal components and linearly reduce the data's dimensionality [26]. Then we model the dataset's PCA representation with t-SNE to obtain plots that model similar sign instances using nearby points with high probability, and dissimilar sign instances using distant points with high probability. The resulting t-SNE plots show clusters that represent which sign instances the model identifies to be similar. However, it is important to note that cluster sizes and distances between clusters are not necessarily meaningful in t-SNE plots. Figure 12 contains 2D embeddings of a set of hotspot images. Individual points are colored according to their ground-truth class. We perform qualitative analysis of sign clusters to understand how the learned representations relate, if at all, to the various ways that scholars have grouped cuneiform signs in the past.

Table 5. Example of tSNE Grouped Signs - Cluster 9

NI	IR	KAL	RU	UN	SA	IB

Table 4 lists the 11 primary sign clusters observed from the 2D t-SNE plots, paired with the predominant signs (labeled using the Elamite sign names devised by [12]), along with comments about the grouping. The full t-SNE plot with associated images is reproduced in Appendix E. Signs which are adjacent in traditional sign lists do appear together in groupings—for example ME and MAŠ in group 1, IGI and KI in group 4 and ḪU and RI in group 7. Groups can also include signs not adjacent or close to each other in traditional sign lists. See for example Table 5 NI, IR, KAL, RU, UN, SA and IB forming group 9. However, all signs do share prominent grouping of two staked horizontals and two to three verticals connected with the lower horizontal. Despite their departures from traditional sign lists, signs are clearly organized by graphic similarity. Further analysis of learned sign representations could therefore assist in developing alternate sign lists for didactic purposes. In addition, our preliminary analysis suggests that the models account for variants in handwriting that would be expected from different scribes writing the same signs. The sign UL appears twice in group 2 and 5 based on the starting wedge which can be (correctly) written as vertical or Winkelhaken. We hypothesize that this direction of analysis could be used to identify individual scribes or scribal schools and group tablets by their author.

5.3 Multistage Pipeline

Previous sections evaluate classification performance with respect to ground-truth bounding boxes, but ultimately the utility of this computer system will be evaluated by its performance on completely unannotated tablets—i.e. where no ground-truth annotations for intermediate pipeline stages are even present. As such, the estimates of classification performance presented earlier are likely upper bounds. We approach this final evaluation in stages: first, by evaluating the performance of the sign classifier on hotspots predicted by the sign detector, where ground-truth labels have been imputed by alignment to ground-truth hotspots, and finally by evaluating the end-to-end character error rate (CER) of transcribed tablets.

Table 6 shows the classification performance of the sign detector trained on ground-truth hotspots, as described in Section 4.3, when performing inference on hotspots predicted by the sign detector described in Section 4.2.

---

[7]Paired with Cameron 1948 indexes

[8]Variant where the first wedge looks vertical.

[9]Variant starting with a vertical rather than a horizontal.

[10]This sign is often perceived as a digraph and is sometimes broken across lines as NU + MAN.

[11]Variant beginning with a Winkelhaken.

Table 4. Observed Sign Groups

Group	Exemplars [7]	Comments
1	BAR (21) NU (89) ME (98) MAŠ (99) SAL (102) MEŠ (109)	Dominant vertical at the beginning of the sign in combination with a few other wedges
2	UL (90) [8] MI (92) [9] SAL (102) MEŠ (109)	Close to group one. Dominant vertical followed by groups of horizontals.
3	MEŠ (109) TUK (112) KU (114) HA (119) EŠŠANA (120) LU (121)	Connected to group 2 via MEŠ. Otherwise, two or more (HA, EŠANNA, LU)) dominant verticals at the beginning of the sign are grouped with serval horizontals, mostly after the verticals.
4	IGI (93) KI (95)	Signs begin with a Winkelhaken followed by a vertical.
5	NUMUN [10] (89) UL (90)[11]	Connected to group 3 via UL and group 4. Signs begin with a Winkelhaken followed by one or more horizontals.
6	IR (58) RU (69) ZI2 (86) EŠ5 = 3 IMIN = 7	Group of three or more (IMIN) dominant verticals. A few IR, RU and ZI2 are included because the grouping of three verticals is dominant in those signs as well.
7	ḪU (30) RI (32)	Both signs have one horizontal followed by two (ḪU) or three (RI) verticals and one Winkelhaken at the end.
8	RA (15) GI (27) IG (28) PI (62) AM (63) DUB (70) MAR (72) UM (73) GAL (83) TUR (84)	All signs in this group do have a combination of the sequence of one to three horizontal(s) – one or two vertical(s) – one to three horizontal(s).
9	NI (57) IR (58) KAL (59) RU (69) UN (75) SA (106) IB (107) and some others	Connected to group 6 via RU and IR. All signs in the group do have a prominent grouping of two staked horizontals and two to three verticals connected with the lower horizontal. Verticals can precede the grouping (SA, IB) or more verticals can be added at the end of the grouping (UN, RU, KAL).
10	ITI (7)	Closely connected with group 9. ITI is distinct from this group as it has a group of one followed by two horizontals and then three Winkelhaken instead of the verticals.
11	SU (85) IŠ (52) DA (77)	Signs begin with two groups of horizontals followed by two to three verticals.

We observe a degradation in performance when evaluating on predicted signs, likely due to some of the issues that we observe with the hotspot detector—erroneous joining or splitting of signs, as well as alignment errors. An end-to-end hotspot detection and labeling training process may provide improved performance here by learning to compensate for shifts in input data distribution induced by the hotspot detection algorithm. To evaluate the CER of transcribed tablets with the current pipeline, we use predicted hotspot locations and top-1 predicted sign labels to predict text ordering using the Sequential RANSAC line detection algorithm. We observe

a relatively high CER, which indicates that the current pipeline is unable to reconstruct text sequences. A top-1 accuracy of approximately 0.56, along with probable alignment errors, is a promising step towards automatic tablet transcription, although insufficient.

Table 6. Classification Performance - Predicted Hotspots

Backbone	Classifier	Top-1 Acc.	Top-3 Acc.	Top-5 Acc.	Detector FPR	CER
ResNet18	ResNet18	0.563 (0.010)	0.735 (0.009)	0.793 (0.009)	0.126 (0.011)	0.683 (0.015)
ResNet18	ResNet50	0.563 (0.010)	0.735 (0.009)	0.793 (0.009)	0.126 (0.011)	0.684 (0.015)
ResNet50	ResNet50	0.540 (0.011)	0.710 (0.009)	0.769 (0.008)	0.121 (0.012)	0.686 (0.023)
ResNet50	ResNet18	0.563 (0.007)	0.734 (0.008)	0.792 (0.008)	0.121 (0.012)	0.669 (0.016)

The combination of each model's pathologies, particularly those of the sign classifier, render the combined pipeline unable to accurately transcribe tablets. This result primarily highlights the need for explicit linguistic supervision and the limits of a vision-only approach to tablet transliteration. The noisy outputs of the sign classifier, while useful for providing suggestions to cuneiformists, most likely require effective denoising uisng linguistic and contextual information.

6 CONCLUSION

6.1 Summary

We demonstrate that the large and richly annotated Persepolis Fortification Archive enables direct training of computer vision models to recognize and classify Elamite cuneiform signs. A trained object detector can localize cuneiform signs with high precision and qualitatively useful performance. A separately-trained classifier is able to produce high-quality predictions of sign identity. These two systems on their own are sufficient to aid cuneiformists in annotating and identifying signs in tablet images, although the current iteration of our end-to-end pipeline has insufficient top-1 sign classification accuracy to produce complete transcriptions automatically without any cuneiformist intervention. The pipeline presented in this work will serve as a baseline for future work on Elamite cuneiform tablet transliteration. We release a processed subset of the Persepolis Fortification Archive dataset as well as the source code and trained model parameters for our pipeline.

6.2 Future Work

While our pipeline can effectively localize cuneiform signs and provide a list of suggested readings, the current system was not built to provide complete transliterations. Future work will seek to rectify this by incorporating linguistic supervision to improve the quality of sign predictions and to provide sign values. This could be achieved in a modular fashion by using a language modeling layer on top of the existing pipeline, or incorporated into the detection phase via a context-aware object detection method such as [15] and [13]. The PFA in OCHRE contains sign to value mappings for each hotspot annotation, so we anticipate this to be feasible on the dataset. However, the latter approach would likely require adopting an end-to-end sign detection and identification scheme rather than the modular scheme we currently adopt.

A second avenue of future work consists of applying the hotspot detector to non-Elamite tablets, to determine whether or not a hotspot detector ostensibly trained without explicit linguistic information can identify signs in other cuneiform corpora. While it is likely that there is some bias towards the specificities of Elamite cuneiform, preliminary experiments (Appendix F) have indicated that the detector component of our pipeline can localize signs on Ur III Sumerian tablets that predate the tablets in the PFA dataset by over a millennium and a half. The method of [20] was shown to perform reasonably well on a different cuneiform corpus, indicating that

there is some transferability between corpora given a trained detector. A combination of our methods and the semi-supervised methods described by [20] may be useful here—for example, using our models (pre-trained on a large annotated dataset) to initialize semi-supervised learning on another cuneiform corpus.

ACKNOWLEDGMENTS

REFERENCES

[1] Roy Abitbol, Ilan Shimshoni, and Jonathan Ben-Dov. 2021. Machine Learning Based Assembly of Fragments of Ancient Papyrus. *J. Comput. Cult. Herit.* 14, 3, Article 33 (July 2021), 21 pages. https://doi.org/10.1145/3460961

[2] Tero Alstola, Shana Zaia, Aleksi Sahala, Heidi Jauhiainen, Saana Svärd, and Krister Lindén. 2019. Aššur and His Friends: A Statistical Analysis of Neo-Assyrian Texts. *Journal of Cuneiform Studies* 71 (Jan. 2019), 159–180. https://doi.org/10.1086/703859

[3] Adam Grant Anderson. 2018. *The Old Assyrian Social Network: an Analysis of the Texts from Kültepe-Kanesh (1950-1750 BCE).* Ph.D. Dissertation. Harvard University.

[4] Yannis M. Assael, Thea Sommerschield, and Jonathan Prag. 2019. Restoring ancient text using deep learning: a case study on Greek epigraphy. *CoRR* abs/1910.06262 (2019). arXiv:1910.06262 http://arxiv.org/abs/1910.06262

[5] Théodore Bluche. 2016. Joint Line Segmentation and Transcription for End-to-End Handwritten Paragraph Recognition. *CoRR* abs/1604.08352 (2016). arXiv:1604.08352 http://arxiv.org/abs/1604.08352

[6] Bartosz Bogacz, Michael Gertz, and Hubert Mara. 2015. Character retrieval of vectorized cuneiform script. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 326–330.

[7] Bartosz Bogacz and Hubert Mara. 2018. From Extraction to Spotting for Cuneiform Script Analysis. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. 199–204. https://doi.org/10.1109/DAS.2018.56

[8] Bartosz Bogacz and Hubert Mara. 2020. Period Classification of 3D Cuneiform Tablets with Geometric Neural Networks. In *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. 246–251. https://doi.org/10.1109/ICFHR2020.2020.00053

[9] Bartosz Bogacz and Hubert Mara. 2022. Digital Assyriology—Advances in Visual Cuneiform Analysis. *J. Comput. Cult. Herit.* 15, 2, Article 38 (may 2022), 22 pages. https://doi.org/10.1145/3491239

[10] Rykle Borger. 2004. *Mesopotamisches Zeichenlexikon.* Vol. 305. Ugarit-Verlag.

[11] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 108–122.

[12] George Glenn Cameron. 1948. Persepolis Treasury Tablets. *Oriental Institute Publications* (1948).

[13] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-End Object Detection with Transformers. *CoRR* abs/2005.12872 (2020). arXiv:2005.12872 https://arxiv.org/abs/2005.12872

[14] Gavin C. Cawley and Nicola L.C. Talbot. 2010. On Over-Fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *J. Mach. Learn. Res.* 11 (Aug. 2010), 2079–2107.

[15] Zhe Chen, Shaoli Huang, and Dacheng Tao. 2018. Context Refinement for Object Detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[16] Jonathan Chung and Thomas Delteil. 2019. A Computationally Efficient Pipeline Approach to Full Page Offline Handwritten Text Recognition. *CoRR* abs/1910.00663 (2019). arXiv:1910.00663 http://arxiv.org/abs/1910.00663

[17] Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. 2009. Power-Law Distributions in Empirical Data. *SIAM Rev.* 51, 4 (2009), 661–703. https://doi.org/10.1137/070710111 arXiv:https://doi.org/10.1137/070710111

[18] Sebastian Colutto, Philip Kahle, Hackl Guenter, and Guenter Muehlberger. 2019. Transkribus. A Platform for Automated Text Recognition and Searching of Historical Documents. In *2019 15th International Conference on eScience (eScience)*. 463–466. https://doi.org/10.1109/eScience.2019.00060

[19] CDLI Contributors. 2022. The Cuneiform Digital Library Initiative. hhttps://cdli.mpiwg-berlin.mpg.de.

[20] Tobias Dencker, Pablo Klinkisch, Stefan M Maul, and Björn Ommer. 2020. Deep learning of cuneiform sign detection with weak supervision using transliteration alignment. *Plos one* 15, 12 (2020), e0243039.

[21] Adinel Dinca and Emil Stetco. 2020. Preliminary Research on Computer-Assisted Transcription of Medieval Scripts in the Latin Alphabet using AI Computer Vision techniques and Machine Learning: A Romanian Exploratory Initiative. *Studia Universitatis Babeș-Bolyai Digitalia* 65, 1 (Dec. 2020), 37–52. https://doi.org/10.24193/subbdigitalia.2020.1.03

[22] Dan Edelstein, Paula Findlen, Giovanna Ceserani, Caroline Winterer, and Nicole Coleman. 2017. Historical Research in a Digital Age: Reflections from the Mapping the Republic of Letters Project. *The American Historical Review* 122, 2 (03 2017), 400–424. https:

//doi.org/10.1093/ahr/122.2.400 arXiv:https://academic.oup.com/ahr/article-pdf/122/2/400/25736565/zah400.pdf

[23] Dietz Otto Edzard and Igor Michajlovich Diakonoff. 1982. Der Aufbau des Syllabars "Proto-Ea". *Societies and languages of the Ancient Near East. Studies in honour of Igor Michailovitch Diakonoff* (1982), 42–61.

[24] D Fisseler, F Weichert, G Müller, and M Cammarosano. 2013. Towards an interactive and automated script feature analysis of 3D scanned cuneiform tablets. *Scientific Computing and Cultural Heritage* (2013), 16.

[25] Center for History and Economics. 2021. *Visualizing Historical Networks.* https://histecon.fas.harvard.edu/visualizing/index.html

[26] Karl Pearson F.R.S. 1901. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2, 11 (1901), 559–572. https://doi.org/10.1080/14786440109462720

[27] Mark B. Garrison and Margaret Cool Root. 2001. *Seals on the Persepolis fortification tablets. Vol. 1, Images of heroic encounter.* The Oriental Institute of the University of Chicago, Chicago.

[28] Shai Gordin, Gai Gutherz, Ariel Elazary, Avital Romach, Enrique Jiménez, Jonathan Berant, and Yoram Cohen. 2020. Reading Akkadian cuneiform using natural language processing. *PloS one* 15, 10 (2020), e0240511.

[29] Alex Graves, Santiago Fernandez, and Juergen Schmidhuber. 2007. Multi-Dimensional Recurrent Neural Networks. https://doi.org/10.48550/ARXIV.0705.2011

[30] Richard T. Hallock. 1969. *Persepolis Fortification Tablets.* Oriental Institute Publications, Vol. 92. University of Chicago Press.

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. https://doi.org/10.48550/ARXIV.1512.03385

[32] Wouter F.M. Henkelman. 2013. 528Administrative Realities: The Persepolis Archives and the Archaeology of the Achaemenid Heartland. In *The Oxford Handbook of Ancient Iran.* Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199733309.013.0019 arXiv:https://academic.oup.com/book/0/chapter/212005924/chapter-ag-pdf/44596812/book_28053_section_212005924.ag.pdf

[33] Tobias Hodel, David Schoch, Christa Schneider, and Jake Purcell. 2021. General models for handwritten text recognition: Feasibility and state-of-the art. German kurrent as an example. *Journal of Open Humanities Data* 7 (Jul 2021). https://doi.org/10.5334/johd.46

[34] Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. 2015. Learning Visual Features from Large Weakly Supervised Data. arXiv:1511.02251 [cs.CV]

[35] Gary King and Langche Zeng. 2001. Logistic Regression in Rare Events Data. *Political Analysis* 9 (2001), 137–163.

[36] Nils M. Kriege, Matthias Fey, Denis Fisseler, Petra Mutzel, and Frank Weichert. 2018. Recognizing Cuneiform Signs Using Graph Based Methods. *CoRR* abs/1802.05908 (2018). arXiv:1802.05908 http://arxiv.org/abs/1802.05908

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (Eds.), Vol. 25. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[38] Yann LeCun and Corinna Cortes. 2010. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/. http://yann.lecun.com/exdb/mnist/

[39] Dar-Shyang Lee and Ray Smith. 2012. Improving book OCR by adaptive language and image models. In *2012 10th IAPR International Workshop on Document Analysis Systems.* IEEE, 115–119.

[40] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. Focal Loss for Dense Object Detection. *CoRR* abs/1708.02002 (2017). arXiv:1708.02002 http://arxiv.org/abs/1708.02002

[41] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2014. Microsoft COCO: Common Objects in Context. https://doi.org/10.48550/ARXIV.1405.0312

[42] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. SSD: Single Shot MultiBox Detector. *Lecture Notes in Computer Science* (2016), 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

[43] Jiaming Luo, Yuan Cao, and Regina Barzilay. 2019. Neural Decipherment via Minimum-Cost Flow: from Ugaritic to Linear B. *CoRR* abs/1906.06718 (2019). arXiv:1906.06718 http://arxiv.org/abs/1906.06718

[44] Jiaming Luo, Frederik Hartmann, Enrico Santus, Yuan Cao, and Regina Barzilay. 2020. Deciphering Undersegmented Ancient Scripts Using Phonetic Prior. arXiv:2010.11054 [cs.CL]

[45] Urs-Viktor Marti and Horst Bunke. 2002. The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition* 5 (2002), 39–46.

[46] Lawrence Mosley. 2013. *A balanced approach to the multi-class imbalance problem.* Ph.D. Dissertation. Iowa State University.

[47] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[48] Steven T Piantadosi. 2014. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review* 21, 5 (2014), 1112–1130.

[49] David MW Powers. 1998. Applications and explanations of Zipf's law. In *New methods in language processing and computational natural language learning*.

[50] Ravneet Punia, Niko Schenk, Christian Chiarcos, and Émilie Pagé-Perron. 2020. Towards the First Machine Translation System for Sumerian Transliterations. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 3454–3460. https://doi.org/10.18653/v1/2020.coling-main.308

[51] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *CoRR* abs/1506.01497 (2015). arXiv:1506.01497 http://arxiv.org/abs/1506.01497

[52] Guillaume Renton, Yann Soullard, Clément Chatelain, Sébastien Adam, Christopher Kermorvant, and Thierry Paquet. 2018. Fully convolutional network with dilated convolutions for handwritten text line segmentation. *International Journal on Document Analysis and Recognition (IJDAR)* 21, 3 (2018), 177–186.

[53] Eugen Rusakov, Kai Brandenbusch, Denis Fisseler, Turna Somel, Gernot A Fink, Frank Weichert, and Gerfrid GW Müller. 2019. Generating Cuneiform Signs with Cycle-Consistent Adversarial Networks. In *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*. 19–24.

[54] Aleksi Sahala, Miikka Silfverberg, Antti Arppe, Krister Lindén, et al. 2020. Automated phonological transcription of Akkadian cuneiform text. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*. European Language Resources Association (ELRA).

[55] Aleksi Sahala, Miikka Silfverberg, Antti Arppe, Krister Lindén, et al. 2020. BabyFST: Towards a finite-state based computational model of ancient Babylonian. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*. European Language Resources Association (ELRA).

[56] Ray Smith. 2007. An overview of the Tesseract OCR engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, Vol. 2. IEEE, 629–633.

[57] Benjamin Snyder, Regina Barzilay, and Kevin Knight. 2010. A Statistical Model for Lost Language Decipherment. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Uppsala, Sweden, 1048–1057. https://aclanthology.org/P10-1107

[58] Matt Stolper. 2007. *Persepolis Fortification Archive Project*. Technical Report. The Oriental Institute at the University of Chicago. https://oi.uchicago.edu/about/annual-reports/oriental-institute-2006-2007-annual-report

[59] Matthew Stolper. 2007. *The Persepolis Fortification Tablets*. Technical Report. The Oriental Institute at the University of Chicago. https://oi.uchicago.edu/sites/oi.uchicago.edu/files/uploads/shared/docs/nn192.pdf

[60] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. *CoRR* abs/1707.02968 (2017). arXiv:1707.02968 http://arxiv.org/abs/1707.02968

[61] Steve Tinney and Eleanor Robson. 2019. About Oracc: Essentials for Oracc users. http://oracc.museum.upenn.edu/doc/about/.

[62] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

[63] E. Vincent and R. Laganiere. 2001. Detecting planar homographies in an image pair. In *ISPA 2001. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces (IEEE Cat.* 182–187. https://doi.org/10.1109/ISPA.2001.938625

[64] Ross Wightman. 2021. Pytorch Image Models (timm). https://fastai.github.io/timmdocs/

[65] Curtis Wigington, Chris Tensmeyer, Brian Davis, William Barrett, Brian Price, and Scott Cohen. 2018. Start, Follow, Read: End-to-End Full-Page Handwriting Recognition. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer International Publishing, Cham, 372–388.

[66] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. 2019. Detectron2. https://github.com/facebookresearch/detectron2.

[67] Tai-Ling Yuan, Zhe Zhu, Kun Xu, Cheng-Jun Li, and Shi-Min Hu. 2018. Chinese Text in the Wild. https://doi.org/10.48550/ARXIV.1803.00085

[68] S. Kevin Zhou, Hayit Greenspan, Christos Davatzikos, James S. Duncan, Bram Van Ginneken, Anant Madabhushi, Jerry L. Prince, Daniel Rueckert, and Ronald M. Summers. 2021. A Review of Deep Learning in Medical Imaging: Imaging Traits, Technology Trends, Case Studies With Progress Highlights, and Future Promises. *Proc. IEEE* 109, 5 (May 2021), 820–838. https://doi.org/10.1109/jproc.2021.3054390

Table 7. Fold Details

Fold	Tablets	Images	Hotspots
0	272	1019	19637
1	272	1112	27819
2	272	1039	21738
3	272	944	23272
4	272	895	23455

A   ADDITIONAL DATA STATISTICS

A.1   Per-Fold Counts

B   LINE DETECTION EXAMPLE OUTPUT
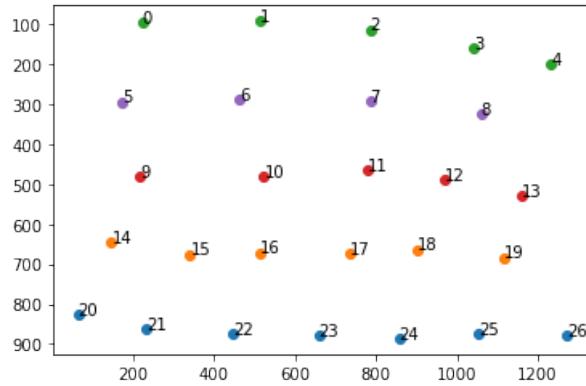
Fig. 13. Example Line Detection Output

Fig. 14. Example of Sequential RANSAC line detection performed on ground-truth hotspot centroids from tablet PF 0008. Detected lines are colored, and predicted sign order is displayed next to the centroid.

C   REWEIGHTING

We attempted to address this per-class imbalance by performing loss reweighting during model training. We experiment with four loss reweighting methods and evaluate their results on top-k accuracy and per-class precision and recall.

We find that simply reweighting according to the frequency of signs in the dataset has a deleterious effect on overall model performance, although it does successfully lower the correlation between training set frequency and test set precision. However, we see a marked increase in the correlation between precision and class frequency. Reweighting according to log-scaled inverse frequencies appears to have little effect on top-k statistics while reducing the correlation between recall and frequency, albeit at the cost of an increase in the correlation between precision and frequency.

Table 8. Reweighting - Top-k, ResNet50

Loss Weighting	Top-1 Acc.	Top-3 Acc.	Top-5 Acc.
Unweighted	0.670 (0.007)	0.843 (0.006)	0.892 (0.005)
Inv. Freq	0.368 (0.038)	0.599 (0.040)	0.698 (0.0348)
Balanced	0.331 (0.023)	0.561 (0.025)	0.663 (0.021)
Log Inv. Freq	0.663 (0.003)	0.840 (0.003)	0.889 (0.003)
Focal	0.671 (0.005)	0.845 (0.004)	0.893 (0.003)

Table 9. Reweighting - Frequency vs Performance, ResNet50

Loss	Mean Recall	Precision ρ	Recall ρ
Unweighted	0.479 (0.019)	0.487 (0.073)	0.767 (0.033)
Inv. Freq	0.304 (0.034)	0.922 (0.011)	0.499 (0.077)
Balanced.	0.264 (0.020)	0.921 (0.014)	0.528 (0.019)
Log Inv. Freq	0.479 (0.011)	0.772 (0.036)	0.609 (0.052)
Focal	0.472 (0.015)	0.477 (0.048)	0.782 (0.050)

D PER-CLASS PRECISION AND RECALL STATISTICS

E FULL TSNE PLOT WITH LABELED POINTS

F GENERALIZATION TO NON-ELAMITE TABLETS

We hypothesized that a cuneiform sign detector trained on Elamite signs would be able to annotate non-Elamite cuneiform tablets, due to general similarities in cuneiform texts across time periods and languages. The shape, style, and even underlying language changed dramatically over the course of cuneiform's 3000 years as a written language, and we wished to see whether or not the signs were similar enough that an object recognition network trained on one period of cuneiform would be able to recognize another period. This also provides some idea of how useful this pipeline component will be as a general cuneiform sign detector, possibly as an initialization for semi- or weakly- supervised methods such as the method described in [20].

Qualitatively, the sign-class-free detector performs fairly well at localizing the cuneiform characters, with issues recognizing signs that are barely visible or that are dissimilar to Elamite cuneiform (upper right). We believe this is a promising indication that a "general" cuneiform sign detector can be produced and adapted to individual script and languages, but much work needs to be done.
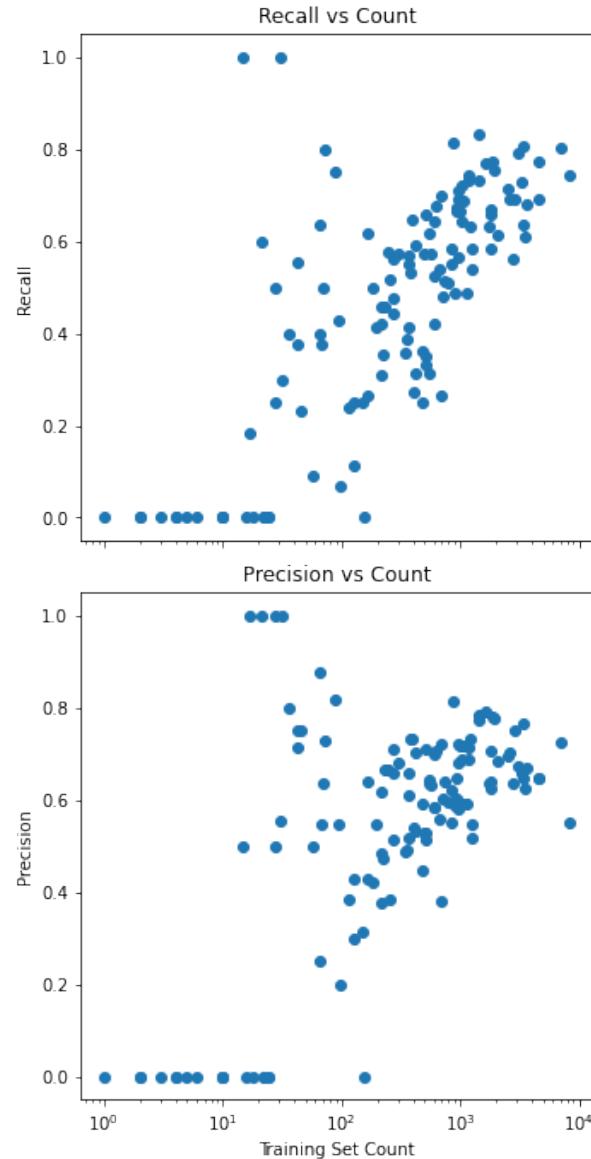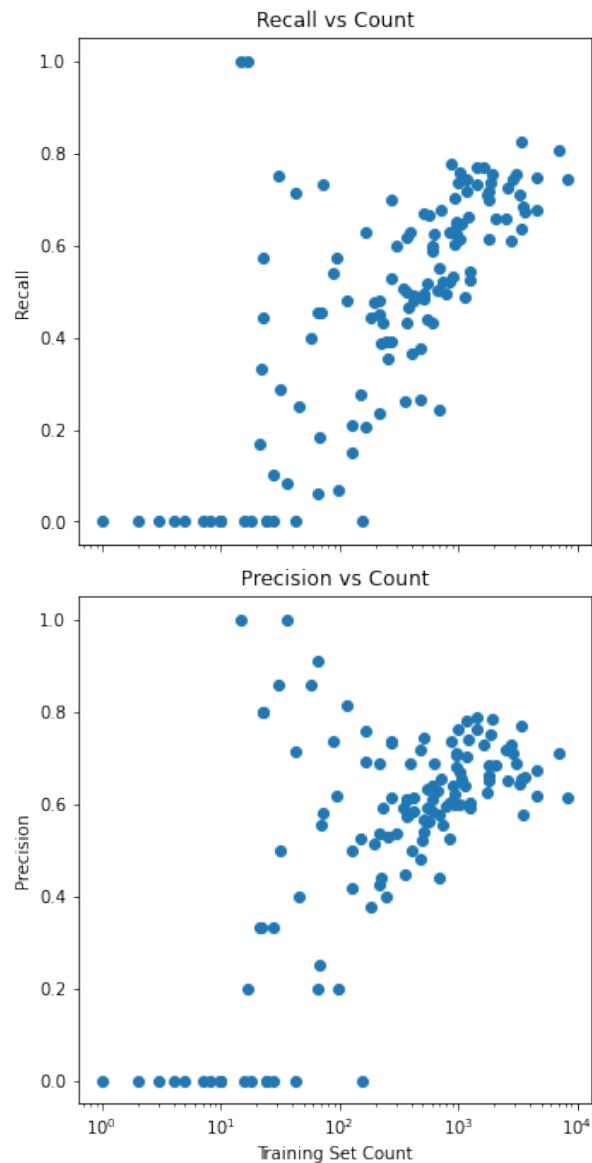
Fig. 15. Per-Class Breakdown - Fold 1

Fig. 16. Per-Class Breakdown - Fold 2

Per-Class Precision and Recall Distributions - Fold 2

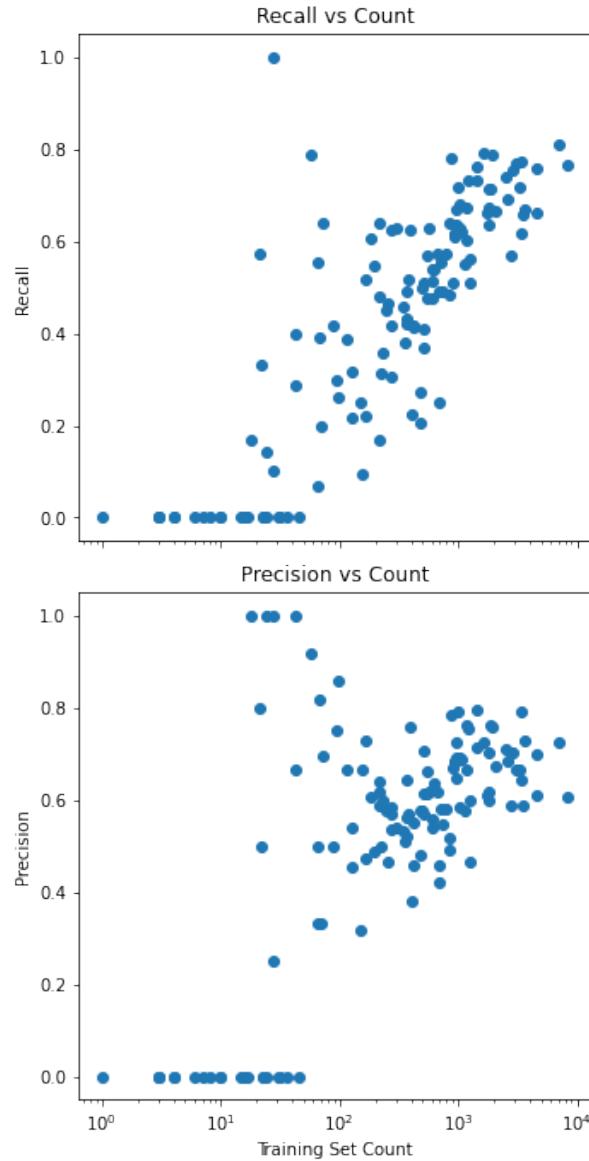Fig. 17. Per-Class Breakdown - Fold 3

Fig. 18. Per-Class Breakdown - Fold 4

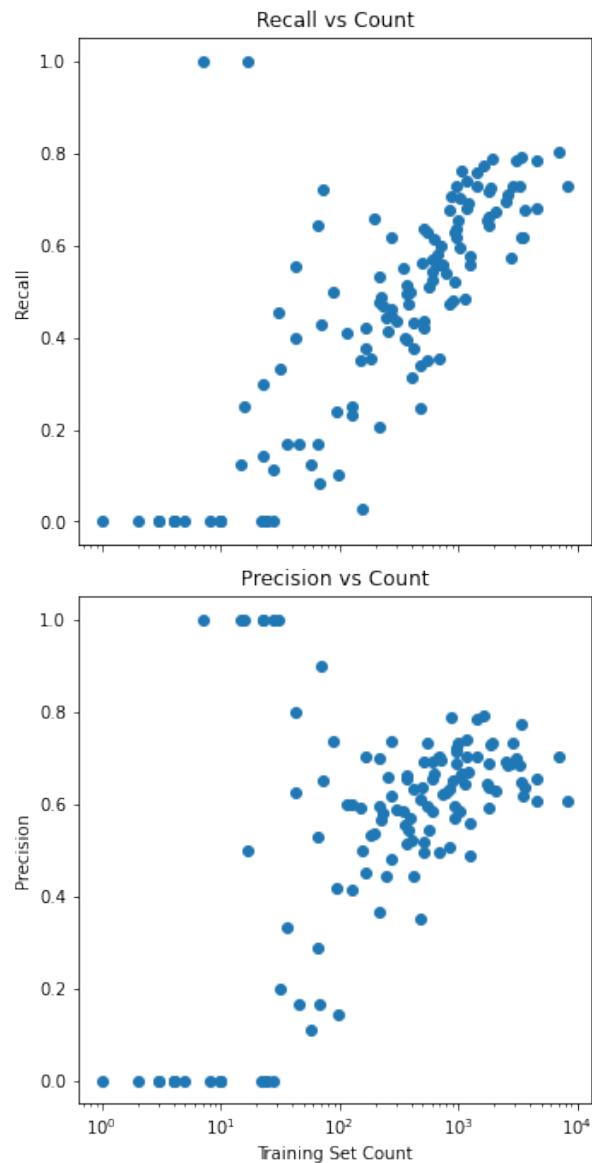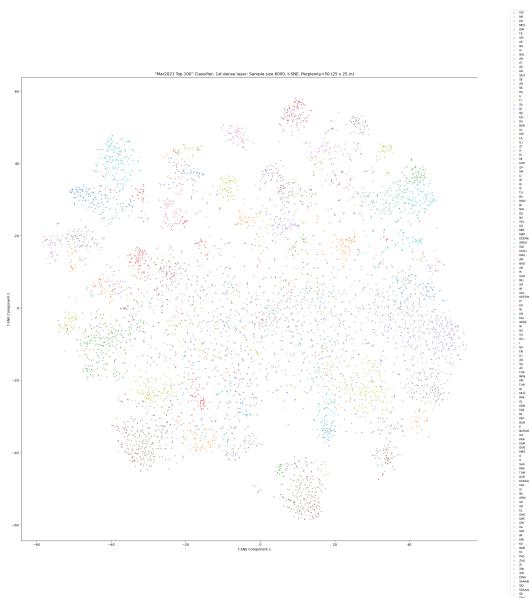Per-Class Precision and Recall Distributions - Fold 4

Fig. 19. Full tSNE Plot With Labeled Points

Fig. 20. Inference on Ur III Tablet